NAMAL UNIVERSITY MIANWALI
DEPARTMENT OF ELECTRICAL ENGINEERING

Communication Systems (Lab)
LAB # 06
REPORT

Title :
Quadrature Amplitude Modulation (QAM) using MATLAB/Simulink

Name	*Fahim Ur Rehman Shah*
Roll No	*NIM-BSEE-2021-24*
Intructor	*Dr. Sajjad Ur Rehman*
Lab Engineer	*Engr. Faizan Ahmad*
Date Performed	*15-April-2024*
Marks	

Introduction

The purpose of this lab is to provide the students basic understanding of DSB-SC Amplitude Modulation and Demodulation in MATLAB/Simulink.

Course Learning Outcomes

CLO2: Develop software simulations to observe the performance of analog and digital communication systems.
CLO4: Report desired results proofs and calculations.

Equipment

- Software
    - MATLAB

Instructions

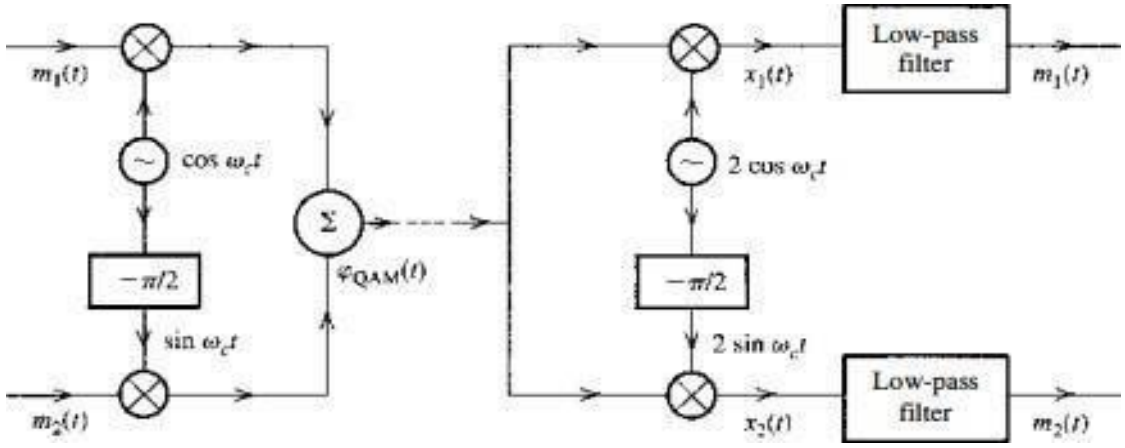
- This is an individual lab. You will perform the tasks individually and submit the required files at the end of the lab.
- Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

Quadrature Amplitude Modulation (QAM):

QAM can be exactly generated without requiring sharp-cut-off band pass filters. QAM operates by transmitting two DSB signals using carriers of the same frequency but in phase quadrature, as shown in Figure below. This scheme is known as quadrature amplitude modulation (QAM) or quadrature multiplexing.

As shown in figure the boxes labelled -pi/2 are phase shifters that delay the phase of an input sinusoid by -n /2 rad. If the two baseband message signals for transmission are m1 (t) and m2 (t), the corresponding QAM signal φQAM (t), the sum of the two DSB-modulated signals, is

$$\varphi_{\mathrm{QAM}}(t) = m_1(t)\cos\ \omega_c t + m_2(t)\sin\ \omega_c t$$



Both modulated signals occupy the same band. Yet two baseband signals can be separated at the receiver by synchronous detection if two local carriers are used in phase quadrature, as shown in Figure.

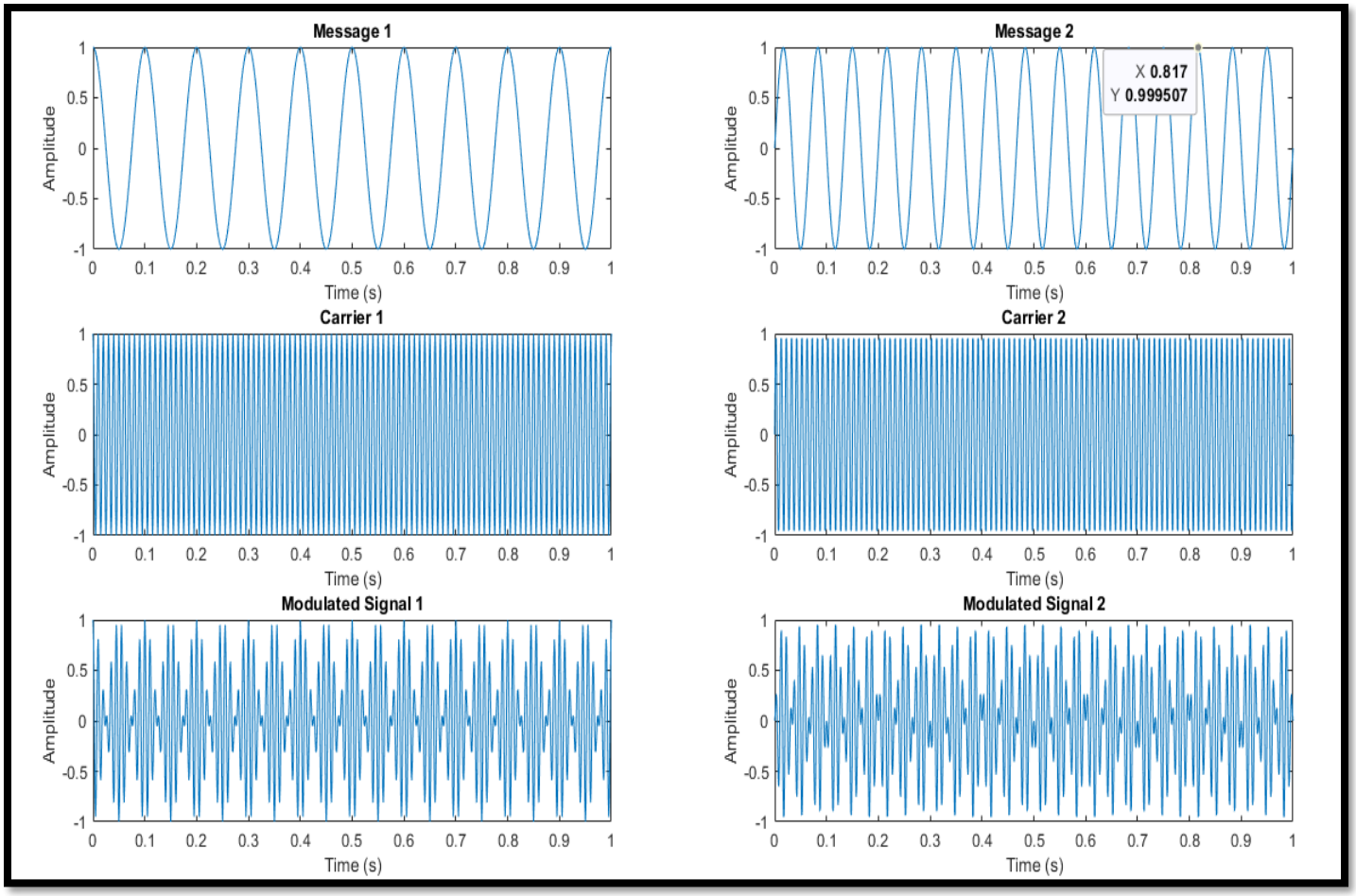
Task 1 – Simple QAM and Demodulation using MATLAB

Exercise 1

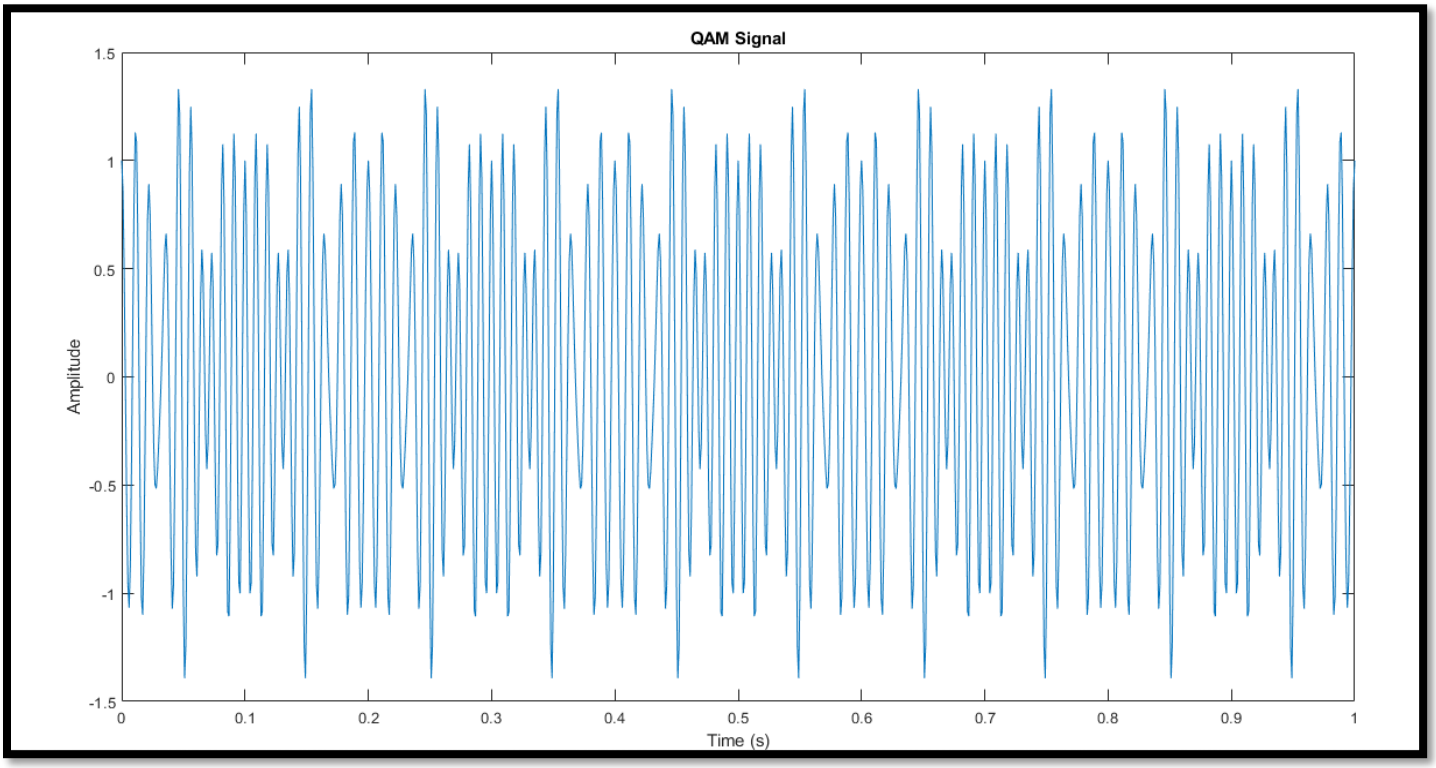
- Here $f_m = 10Hz$, $f_{m1} = 15Hz$, time vector 0:1/ f_s:1, f_s =1000Hz and $f_c = 100Hz$.
- Define message 1 as cos (2*pi* f_m *t)
- Define message 2 as sin (2*pi* f_{m1} *t)
- Take carrier 1 as cos (2*pi* f_c *t).
- Take carrier 2 as sin(2*pi* f_c *t)
- Perform AM modulation of the both signals by multiplying the both messages with its respective carrier.
- Add both modulated signals for QAM and plot the results using plot command.

```
% Given parameters
fm = 10; % Hz
fm1 = 15; % Hz
fs = 1000; % Hz
fc = 100; % Hz

% Define time vector
t = 0:1/fs:1;

% Define message 1 as cos(2*pi*fm*t) and plot
message1 = cos(2*pi*fm*t);
subplot(3, 2, 1);
plot(t, message1);
title('Message 1');
xlabel('Time (s)');
ylabel('Amplitude');

% Define message 2 as sin(2*pi*fm1*t) and plot
message2 = sin(2*pi*fm1*t);
subplot(3, 2, 2);
plot(t, message2);
title('Message 2');
xlabel('Time (s)');
ylabel('Amplitude');

% Take carrier 1 as cos(2*pi*fc*t) and plot
carrier1 = cos(2*pi*fc*t);
subplot(3, 2, 3);
plot(t, carrier1);
title('Carrier 1');
xlabel('Time (s)');
ylabel('Amplitude');

% Take carrier 2 as sin(2*pi*fc*t) and plot
```

```
carrier2 = sin(2*pi*fc*t);
subplot(3, 2, 4);
plot(t, carrier2);
title('Carrier 2');
xlabel('Time (s)');
ylabel('Amplitude');

% Perform AM modulation of the both signals by multiplying the both messages with its
respective carrier
modulated_signal1 = message1 .* carrier1;
modulated_signal2 = message2 .* carrier2;

% Plot modulated signals
subplot(3, 2, 5);
plot(t, modulated_signal1);
title('Modulated Signal 1');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(3, 2, 6);
plot(t, modulated_signal2);
title('Modulated Signal 2');
xlabel('Time (s)');
ylabel('Amplitude');
```



```
% Add both modulated signals for QAM
qam_signal = modulated_signal1 + modulated_signal2;

% Plot the result of QAM modulation
figure;
plot(t, qam_signal);
title('QAM Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```



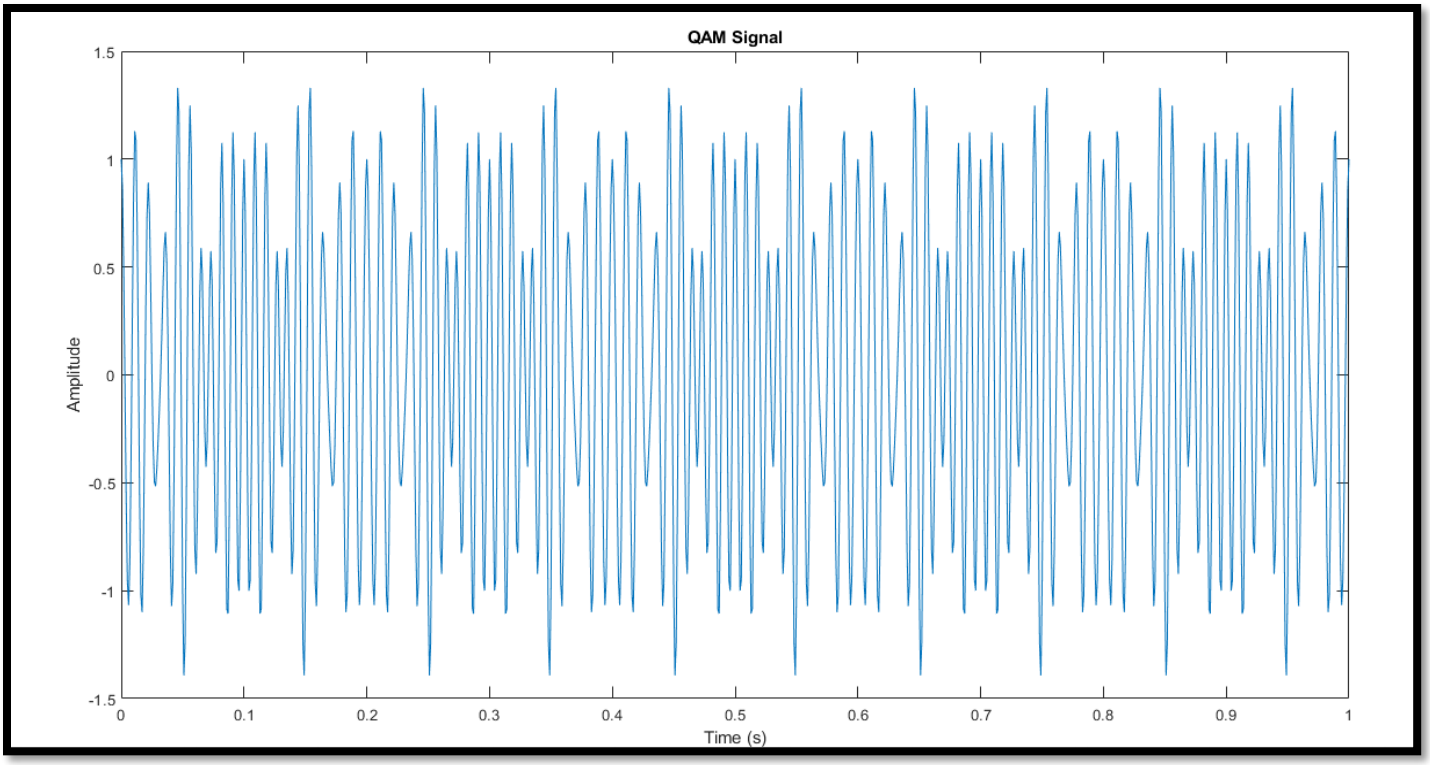
Explanation:

In this lab task for Communication Systems, I was given parameters like the frequency of the message signals (fm and fm1), the sampling frequency (fs), and the carrier frequency (fc). I then defined two message signals, message1 and message2, by using cosine and sine functions with the given frequencies. After that, I created two carrier signals, carrier1 and carrier2, using cosine and sine functions with the carrier frequency. To perform Amplitude Modulation (AM), I multiplied each message signal with its respective carrier signal to get modulated_signal1 and modulated_signal2. By adding these modulated signals together, I obtained a Quadrature Amplitude Modulation (QAM) signal, qam_signal, which was then plotted to visualize the result of the modulation process.

Exercise 2

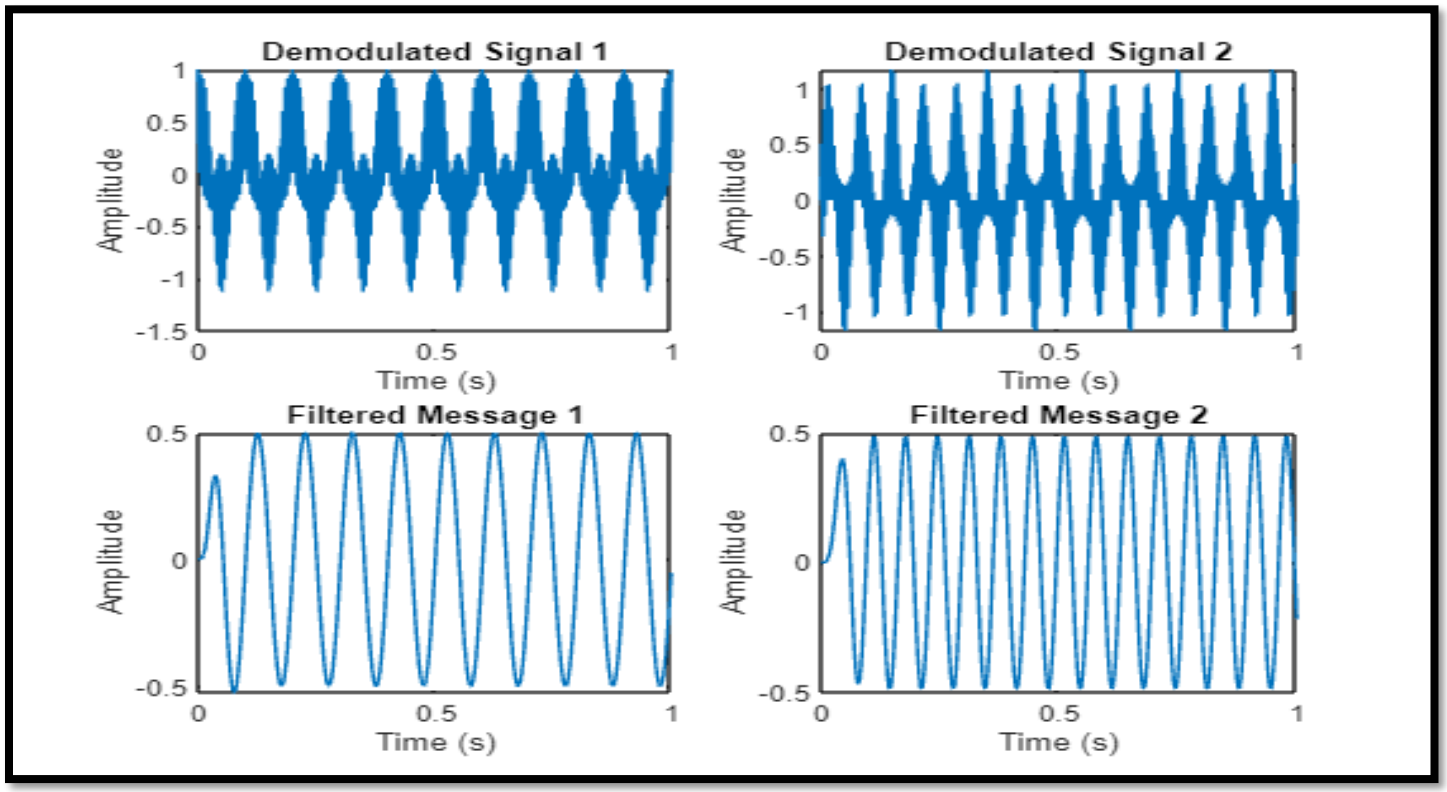
- Multiply the QAM signal with respective carriers for QAM demodulation.
- Plot all the signals in 4x2 figures using plot and sub plot commands by giving proper titles.
- Design Low Pass butter filter according to the message signals requirements.

```
% Given parameters
fm = 10; % Hz
fm1 = 15; % Hz
fs = 1000; % Hz
fc = 100; % Hz

% Define time vector
t = 0:1/fs:1;

% Define message 1 as cos(2*pi*fm*t) and plot
message1 = cos(2*pi*fm*t);

% Define message 2 as sin(2*pi*fm1*t) and plot
message2 = sin(2*pi*fm1*t);

% Take carrier 1 as cos(2*pi*fc*t) and plot
carrier1 = cos(2*pi*fc*t);

% Take carrier 2 as sin(2*pi*fc*t) and plot
carrier2 = sin(2*pi*fc*t);

% Perform AM modulation of the both signals by multiplying the both messages with its
respective carrier
modulated_signal1 = message1 .* carrier1;
modulated_signal2 = message2 .* carrier2;

% Add both modulated signals for QAM
qam_signal = modulated_signal1 + modulated_signal2;
% Plot the result of QAM modulation
figure;
plot(t, qam_signal);
title('QAM Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```




```
% Multiply QAM signal with respective carriers for QAM demodulation
demod_signal1 = qam_signal .* carrier1;
demod_signal2 = qam_signal .* carrier2;

% Plot demodulated signal 2
subplot(2, 2, 1);
plot(t, demod_signal1);
title('Demodulated Signal 1');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot demodulated signal 2
subplot(2, 2, 2);
plot(t, demod_signal2);
title('Demodulated Signal 2');
xlabel('Time (s)');
ylabel('Amplitude');

% Design Low Pass Butterworth filter for message signals requirements
% Define filter parameters
cutoff_frequency = 20; % Hz
order = 5; % Filter order
[b, a] = butter(order, cutoff_frequency/(fs/2), 'low');

% Apply filter to message signals
filtered_message1 = filter(b, a, demod_signal1);
filtered_message2 = filter(b, a, demod_signal2);

% Plot filtered messages
```

```
subplot(2, 2, 3);
plot(t, filtered_message1);
title('Filtered Message 1');
xlabel('Time (s)');
ylabel('Amplitude');
subplot(2, 2, 4);
plot(t, filtered_message2);
title('Filtered Message 2');
xlabel('Time (s)');
ylabel('Amplitude');
```



Explanation:

In this Communication Systems lab task, I worked with message signals (message1 and message2) and carrier signals (carrier1 and carrier2) defined by specific frequencies. I performed Amplitude Modulation (AM) by multiplying each message with its respective carrier to get modulated signals. These modulated signals were then added to create a Quadrature Amplitude Modulation (QAM) signal. Next, I multiplied the QAM signal with the carriers again for QAM demodulation, resulting in demod_signal1 and demod_signal2. To meet the message signals' requirements, I designed a Low Pass Butterworth filter with a specified cutoff frequency and filter order. Applying this filter to the demodulated signals produced filtered_message1 and filtered_message2, which were then plotted to visualize the filtered messages.

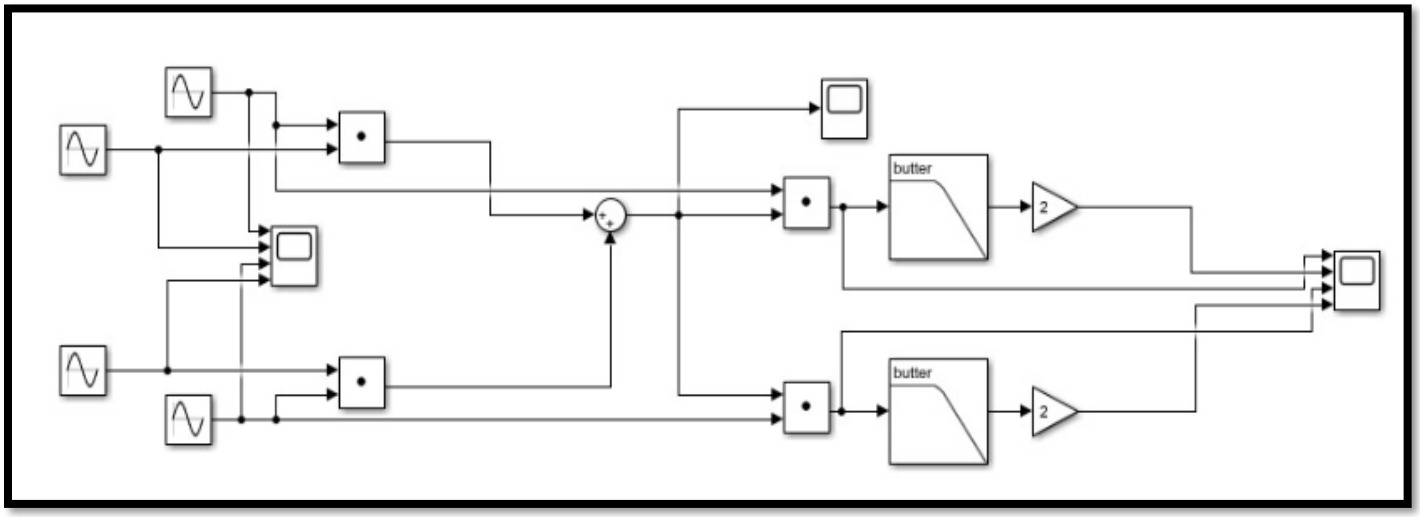
Task 2 – QAM and Demodulation using Simulink

Exercise 3

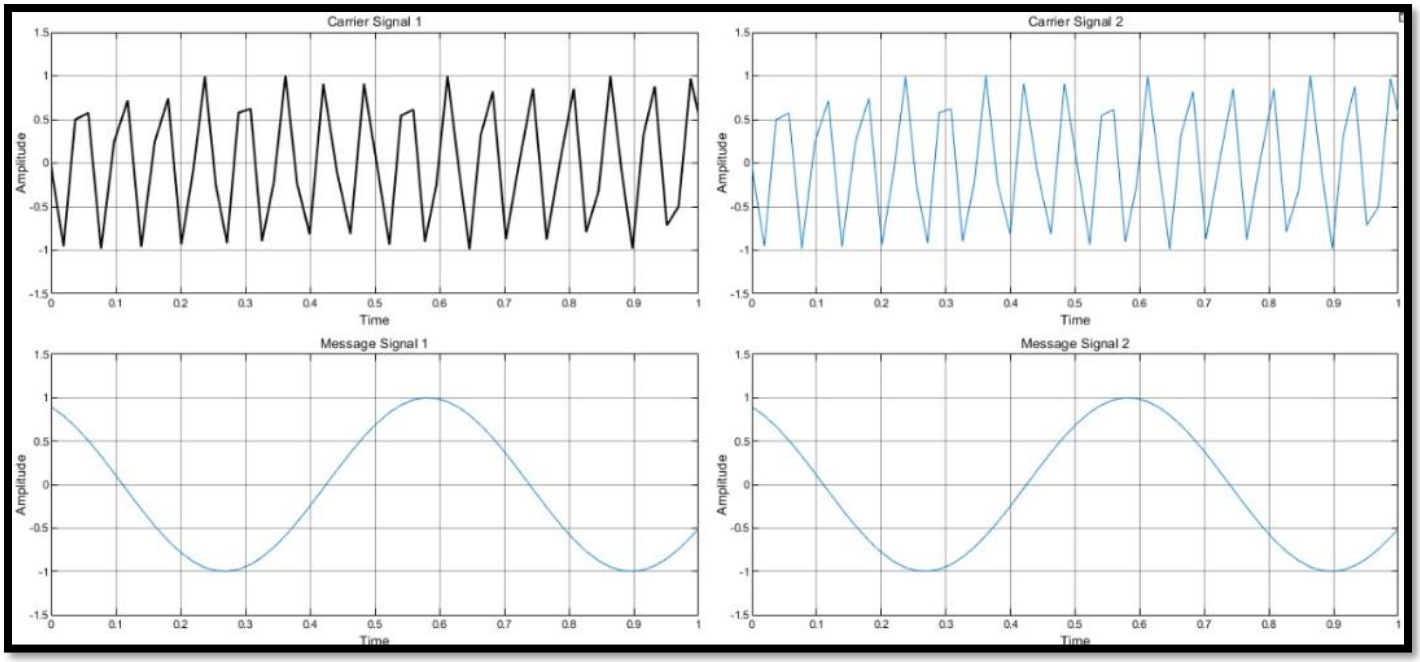
Implement QAM Modulation and De-modulation using Simulink, you can use the following blocks in Simulink to implement it. Where $f_m = 10Hz$ and $f_c = 100Hz$

- Sine Wave (Carrier and Message Generation)
- Dot Product (for multiplication)
- Analog Filter Design (Butter-low pass)
- Gain (use at Demodulation side)
- Scope

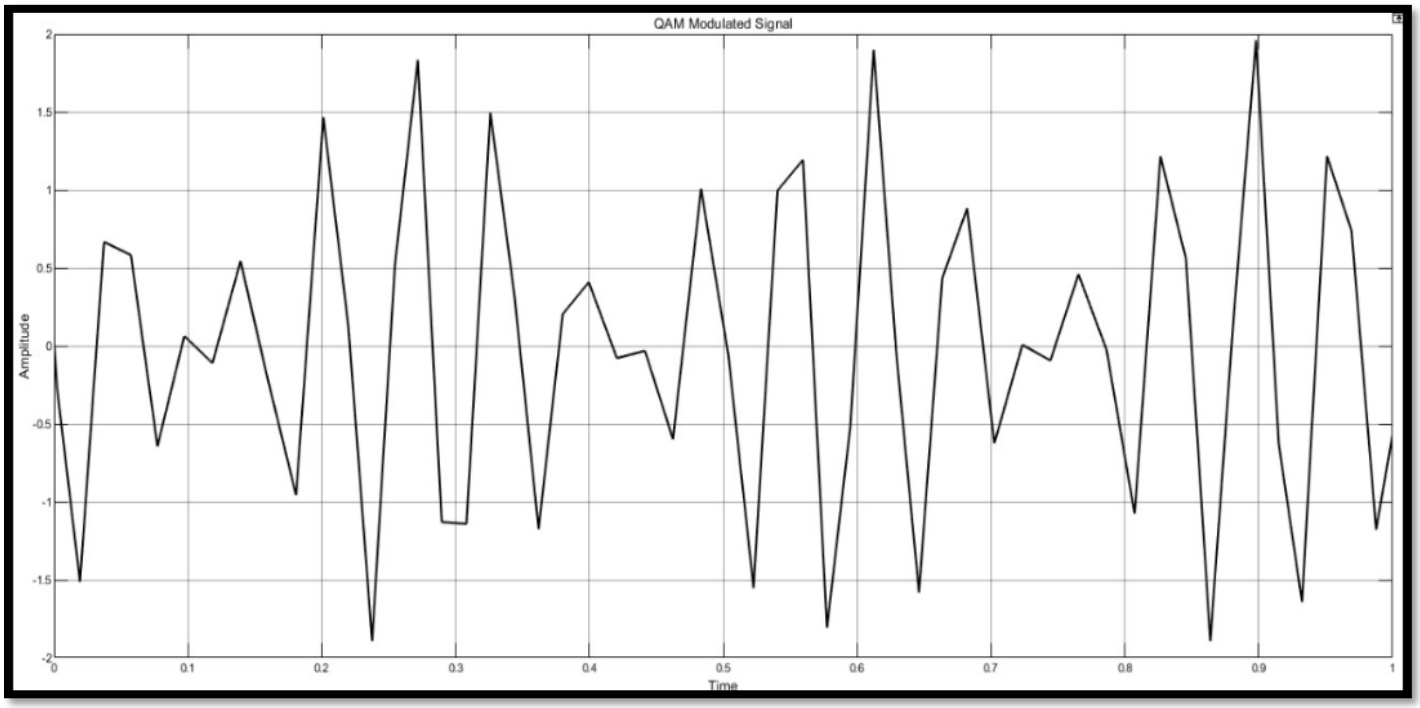
Simulink Diagram:



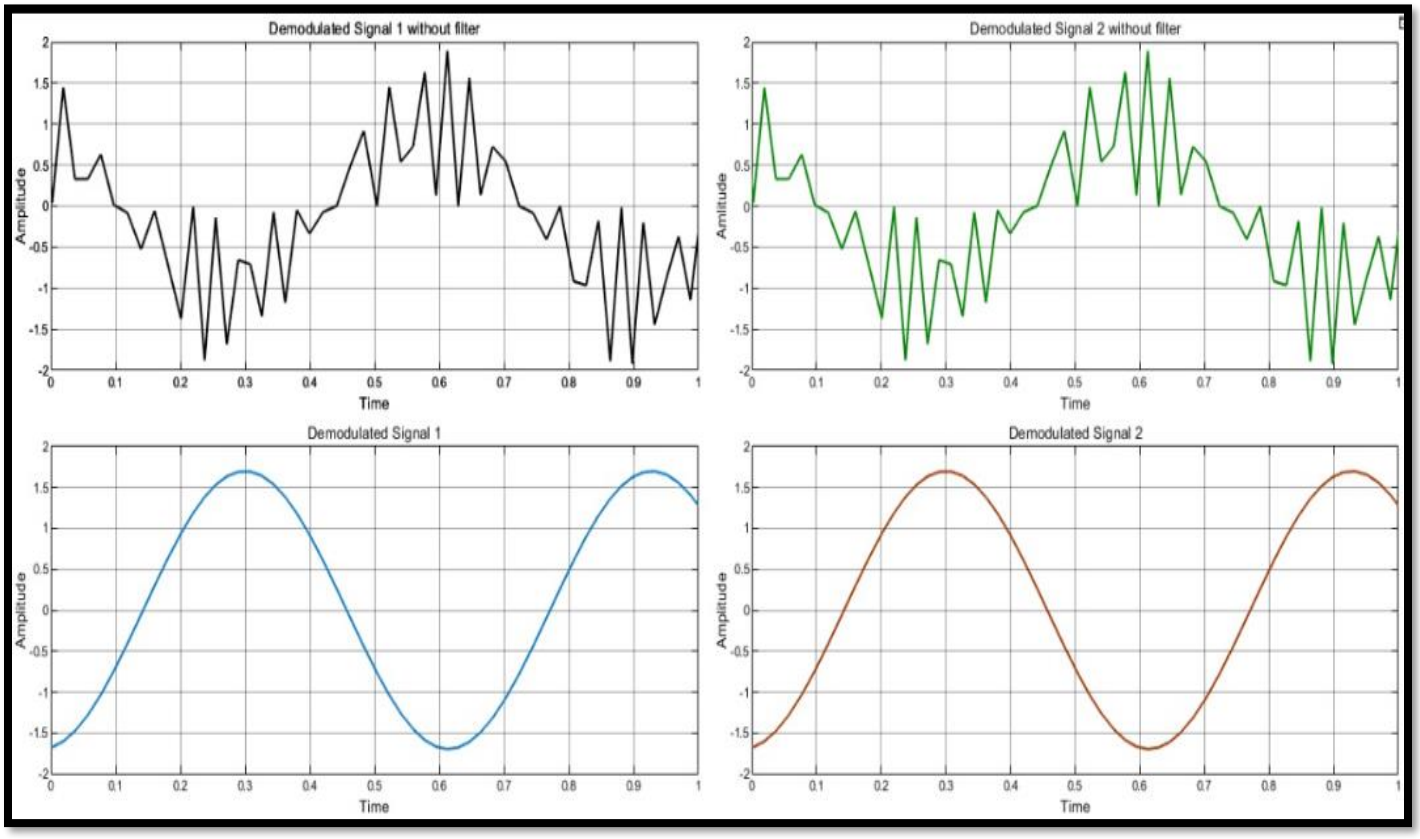
Message and Carrier Signal Output:



QAM Modulated Signal Output:



Demodulated Signal Output:



Explanation:

To implement Quadrature Amplitude Modulation (QAM) Modulation and Demodulation using Simulink, I started by using the Sine Wave block to generate the carrier signal and message signal with frequencies $f_m = 10Hz$ and $f_c = 100Hz$. Then, I used the Dot Product block to perform the multiplication of the message signal with the carrier signal for Modulation. Next, I designed an Analog Low Pass Butterworth Filter using the Analog Filter Design block to filter the demodulated signal. Additionally, I incorporated a Gain block at the Demodulation side to adjust the signal strength. Finally, I used the Scope block to visualize and analyze the signals at different stages of the Modulation and Demodulation process. By setting up these blocks and connecting them appropriately in Simulink, I was able to simulate and observe the QAM Modulation and Demodulation process visually.

Conclusion:

In conclusion, I successfully implemented Quadrature Amplitude Modulation (QAM) Modulation and Demodulation using Simulink. By generating the carrier and message signals, performing the multiplication for modulation, designing an Analog Low Pass Butterworth Filter for signal filtering, adjusting signal strength with a Gain block, and visualizing the process using the Scope block, I was able to simulate and observe the QAM Modulation and Demodulation process effectively. This hands-on experience provided valuable insights into signal processing techniques and communication systems within the context of Simulink simulation, enhancing my understanding of practical applications in the field.

Com. Sys. Lab 6 Rubric

Method of Evaluation: Executable code, Report submitted by students **Measured**
Learning Outcomes:
CLO2: Develop software simulations to observe the performance of analog and digital communications systems.
CLO4: Report desired results proofs and calculations.

	Excellent 10	Good 9-7	Satisfactory 6-4	Unsatisfactory 3-1	Poor 0	Marks Obtained
Code (CLO2)	Correct code, easily understandable with comments where necessary	Correct code but without proper indentation or comments	Slightly incorrect code with proper comments	Incorrect code with improper format and no comments	Code not submitted	
Output (CLO2)	Output correctly shown with all Figures/ Plots displayed as required and properly labelled	Most Output/ Figures/ Plots displayed with proper labels	Some Output/ Figures/ Plots displayed with proper labels OR Most Output/ Figures/ Plots displayed but without proper labels	Most of the required Output/ Figures/ Plots not displayed	Output/ Figures/ Plots not displayed	
Answers (CLO2)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Wrong Answers	
Lab Report (CLO4)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/ punctuation	Report not submitted	
Total						